IN THE SPECIFICATION

Please amend the paragraphs beginning at page 12, line 11 as follows:

The example shown in Figs. 4a-4d is based on the following three illustrative documents that are not shown in Fig. 4.:

file "soup 1.doc" contains the ingredients of soup no. 1 and reads "tomatoes, water, paprika, salt, and flavours"

file "soup 2.doc" contains the ingredients of soup no. 2 and reads "tomatoes, onions, potatoes, water, ~~paprika,~~ salt, and flavours";

file "soup 3.doc" contains the ingredients of soup no. 3 and reads "tomatoes, onions, potatoes, water, pork, <u>and</u> salt~~, and flavours~~".

All words contained in the three documents are listed in the word-list with the following conditions:

a) they are usually ordered alphabetically;

b) they are usually presented in their "normal-form" like nominative singular for nouns, or infinitive for verbs, respectively. This means that if in case of a different example the words "mouse" and "mice" would both be relevant for the word-list only the word "mouse" would be shown: if the user should select the word "mouse" from the word-list, another list not shown in this example might pop up showing the two entries "mouse" and "mice";

c) usually only relevant words and operations are being listed. In this example the word "and" is contained in the documents, but since it bears no information for the user it is being suppressed.

No word ~~and no operation~~ has been selected yet in Fig. 4a.

If the soup should contain onions, the word "onion" is chosen. Thus follows; since soup 1 does not contain onions, it <u>does not show up in</u> ~~is omitted from~~ the result-list. ~~Thus soup 1 shown in Fig. 4a is removed yielding the result-list~~ in Fig. 4b. The result list shows soup 2 and soup 3, which means that they and only they contain onions.

<u>The operator-list is updated to display the operators "AND", "NOT", "OR" and "EXCL. OR", as the word "onion" when combined with any one of these four operators would produce at least one result for the result-list.</u>

So far everything is temporary and the user can still change his mind and move the focus from "onions" to another item; when scrolling through the word-list the values <u>in the operator-list and</u> in the result-list and the entry in the history-list would change correspondingly and more or less simultaneously; since everything is only temporary the word list still contains all words including those from the file "soup 1".

<u>If the user were to choose "tomato" instead of "onion", the operator "EXCL. OR" would have to be taken off the operator-list. Since each soup contains tomatoes, there would be no result fulfilling the operation "tomato EXCL. OR ...".</u>

<u>As soon as the user is sure that it is onions he wants he fixes the word "onion" as the first datum of the operation.</u>

After having chosen "onions" the user decides that the soup should contain onions, but no pork<u>, and therefore selects the operator "NOT".</u> ~~This is accomplished in 2 steps:~~

~~First the operator "NOT" is chosen as in Fig. 4c. The moment the operator is chosen the selection of the word "onion" becomes fixed as the first datum of the operation.~~

This makes it possible to update the word-list <u>as shown in FIG. 4c:</u> ~~; it is now known that only those documents are to be selected which contain onions; these are the documents "soup 2"~~

~~and "soup 3". So from now on the word-list should show only those words that are contained in these two documents—all other words must be suppressed in order to avoid null values. This omits "paprika" from the word-list, since paprika is contained only in soup 1, and ensures that none of the remaining items in the word-list produces a null-value.~~

- the word "paprika" is eliminated because it is contained neither in "soup 2" nor in "soup 3";

- the words "onion", "potato", "salt", "tomato" and "water" are eliminated because they are present in "soup 2" and "soup 3", and are therefore incapable of satisfying the "NOT" condition

Accordingly, only the words "flavour" and "pork" are left.

If for example ~~"paprika"~~ "water" would not be taken off the word-list, null values could occur; this would be the case if the user would select "onion~~" AND "~~ NOT water ~~paprika~~", since ~~there is no~~ each soup containing ~~both~~ onions also contains water ~~and paprika~~.

So far only the first datum of the operation is fixed; the ~~operation~~ operator itself is still temporary and the user may change his mind and move the focus from "NOT" to another ~~operation~~ operator. When scrolling through the ~~operation~~ operator-list the ~~entry~~ entries in the word-list and in the history-list would change correspondingly and more or less simultaneously.

As soon as the user is sure that it is the operator "NOT" that is wanted, he/she fixes it as the first operator of the operation.

So far, a word and an operator have been chosen. Now another word is needed to complete the operation. As it is decided that the soup should not contain "pork", ~~Then~~ "pork" is selected in the word-list as shown in Fig. 4d.

"Soup 3" must be omitted from the result-list because it contains pork, so "soup 2" becomes the only entry in the result-list because it is the only result containing "onions" but no "pork".

The operators "NOT" and "EXCL. OR" must be omitted from the operator-list because there is no word in the word-list that can fulfill the operation "(onions NOT pork) NOT ..." or "(onions NOT pork) EXCL. OR ...".

At this point, as a word has not been fixed, the user may scroll from "pork" to "flavour". When he/she scrolls through the word-list, the values in the operator-list, the result-list and history-list will change correspondingly and simultaneously.

As soon as the user fixes the word "pork", a single result ("soup 2") is obtained and the query is essentially over. It would be possible to continue with the operators "AND or "OR", but this would not make much sense as the result "soup 2" would not change.

~~The moment the second datum (in this case "pork") is chosen the selection of the operation "NOT" becomes fixed as the operator. So the operation itself is on one hand finished, since it reads "onion NOT pork", which makes it complete and valid with only "soup2" being left in the result-list.~~

~~On the other hand it is still temporary and the user may change his mind and move the focus from "pork" to another item. When scrolling through the word-list the entries in the operation-list, in the result-list and in the history-list would change correspondingly and more or less simultaneously.~~

Since all unfitting items ~~(in this example "paprika") had been~~ <u>were</u> omitted from the word-list the moment the operator (in this example "NOT") was chosen it is ensured that null values cannot occur ~~no matter~~ <u>regardless of</u> which item the user should choose as the second word ~~one~~ (in this example "pork")~~;~~ <u>As a result,</u> there will always be at least be one document in the result-list ~~(in this example it will either be one document or two; three are no longer possible since document "soup 1" has been eliminated)~~.

So in a certain sense the operation is never permanently fixed since the result is shown in the result list while the user is still scrolling through the word list choosing his second datum. If desired the operation can be defined as fixed by clicking on an OK-button.

The operation reads "onion NOT pork" and is complete, because only one single document is left in the result-list. Even though it makes no sense to continue it is not forbidden to do so and to expand the operation for example to "(onion NOT pork) AND <u>flavour</u> ~~potato~~" which would result in the same "soup 2". This would be accomplished by selecting the operator "AND" ~~(which would leave only those words in the word-list that are contained in the document "soup 2")~~, and then the word "~~potato~~ <u>flavour</u>."